

# Dialog Elements

---

## ActiveX Controls



This is an example of an ActiveX control.

### Description

ActiveX controls are third-party custom controls that you can integrate into a Natural dialog. In addition to the Natural attributes listed below, the properties, methods and events of the respective ActiveX control are available. The handling of these properties, methods and events is described in the User's Guide - Event driven programming techniques - Working with ActiveX controls..

### Natural Attributes for ActiveX Controls

Attribute Name	Query	Set/Modify	In Attr. Window
CLIENT-DATA	X	X/X	
CLIENT-KEY	X	X/X	
CLIENT-VALUE	X	X/X	
CONTEXT-MENU	X	X/X	X
DIL-TEXT	X	X/X	X
ENABLED	X	X/X	X
FIRST-CHILD	X	-/-	
FOLLOWS	X	X/X	
HELP-ID	X	X/X	X
LAST-CHILD	X	-/-	
PARENT	X	X/-	
PREDECESSOR	X	-/-	
RECTANGLE-H	X	X/X	X
RECTANGLE-W	X	X/X	X
RECTANGLE-X	X	X/X	X
RECTANGLE-Y	X	X/X	X
STYLE	X	X/X	X
SUCCESSOR	X	-/-	
TYPE	X	X/-	
VISIBLE	X	X/X	X

## Events

This dialog element does not create Natural-specific events but generates Natural events from the events available with the ActiveX control.

## Bitmap Control



### Description

A bitmap control is a picture to be displayed anywhere within a dialog window. It helps explain the purpose of dialog elements in a graphical way. The end user may click upon a bitmap control and drag it onto another bitmap control in the same dialog. You can allow this by using the attributes `ENABLED` and `DRAGGABLE` and the `PROCESS GUI` statement actions `INQ-CLICKPOSITION` and `INQ-DRAG-DROP`.

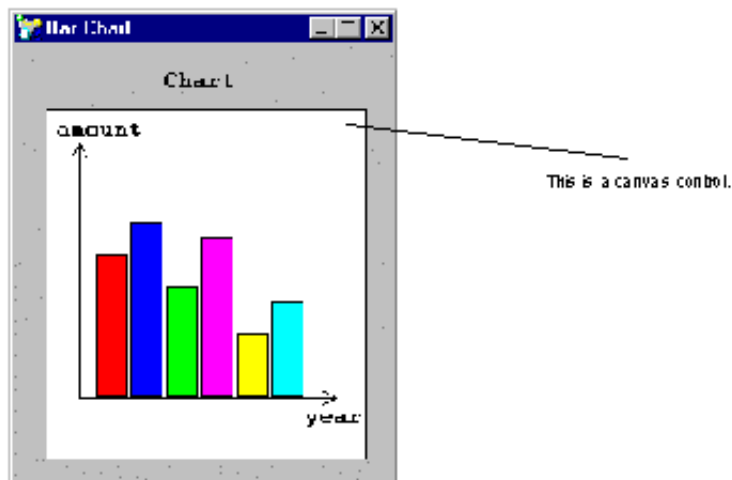
### Attributes for Bitmap Control

Attribute Name	Query	Set/Modify	In Attr. Window
ACCELERATOR	X	X/X	X
BITMAP-FILE-NAME	X	X/X	X
CLIENT-DATA	X	X/X	
CLIENT-HANDLE	X	X/X	
CLIENT-KEY	X	X/X	
CLIENT-VALUE	X	X/X	
CONTEXT-MENU	X	X/X	X
DIL-TEXT	X	X/X	X
DRAGGABLE	X	X/X	X
ENABLED	X	X/X	X
FIRST-CHILD	X	-/-	
FOLLOWS	X	X/X	
HELP-ID	X	X/X	X
LAST-CHILD	X	-/-	
PARENT	X	X/-	
PREDECESSOR	X	-/-	
RECTANGLE-H	X	X/X	X
RECTANGLE-W	X	X/X	X
RECTANGLE-X	X	X/X	X
RECTANGLE-Y	X	X/X	X
STYLE	X	X/X	X
SUCCESSOR	X	-/-	
SUPPRESS-CLICK-EVENT	X	X/X	
SUPPRESS-DBL-CLICK-EVENT	X	X/X	
TYPE	X	X/-	
VISIBLE	X	X/X	X

## Events

Click event (may be suppressed), double-click event (may be suppressed), drag and drop event.

## Canvas Control



### Description

A canvas control provides a necessary background for the rectangle, line and graphic-text controls. Once you have created a canvas control in the dialog, you can go on to create the rectangle, line and graphic-text controls in it. The rectangle, line and graphic-text controls are then displayed inside the borders of the canvas control; if they go beyond the canvas borders, they are clipped.

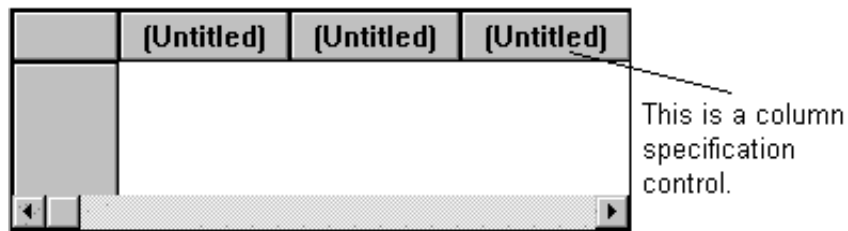
### Attributes for Canvas Control

Attribute Name	Query	Set/Modify	In Attr. Window
BACKGROUND-COLOUR-NAME	X	X/X	X
BACKGROUND-COLOUR-VALUE	X	X/X	X
CLIENT-DATA	X	X/X	
CLIENT-HANDLE	X	X/X	
CLIENT-KEY	X	X/X	
CLIENT-VALUE	X	X/X	
CONTEXT-MENU	X	X/X	X
DIL-TEXT	X	X/X	X
ENABLED	X	X/X	X
FIRST-CHILD	X	-/-	
FOLLOWS	X	X/X	
FOREGROUND-COLOUR-NAME	X	X/X	X
FOREGROUND-COLOUR-VALUE	X	X/X	X
HELP-ID	X	X/X	X
LAST-CHILD	X	-/-	
OFFSET-X	X	X/X	
OFFSET-Y	X	X/X	
PARENT	X	X/-	
PREDECESSOR	X	-/-	
RECTANGLE-H	X	X/X	X
RECTANGLE-W	X	X/X	X
RECTANGLE-X	X	X/X	X
RECTANGLE-Y	X	X/X	X
STYLE	X	X/-	X
SUCCESSOR	X	-/-	
SUPPRESS-CHANGE-EVENT	X	-/-	
SUPPRESS-CLICK-EVENT	X	X/X	
SUPPRESS-DBL-CLICK-EVENT	X	X/X	
TYPE	X	X/-	
VISIBLE	X	X/X	X

## Events

Click event (may be suppressed), double-click event (may be suppressed).

## Column-Specification Control



### Description

A column-specification control is a dialog element that defines the columns in a table control. (A table control is a dialog element that represents a spreadsheet.) Once the table control defines the spreadsheet as such, its columns are defined by adding column-specification controls.

### Attributes for Column-Specification Control

Attribute Name	Query	Set/Modify	In Attr. Window
CLIENT-DATA	X	X/X	
CLIENT-HANDLE	X	X/X	
CLIENT-KEY	X	X/X	
CLIENT-VALUE	X	X/X	
COLUMN-TYPE	X	X/-	X
DIL-TEXT	X	X/X	X
FIRST-CHILD	X	-/-	
HELP-ID	X	X/X	X
LAST-CHILD	X	-/-	
LENGTH	X	X/-	X
MODIFIABLE (input-field)	X	X/X	X
MODIFIABLE (selection-box)	X	X/-	X
PARENT	X	X/-	
PREDECESSOR	X	-/-	
RECTANGLE-W	X	X/X	X
STRING	X	X/X	X
STYLE	X	X/-	
SUCCESSOR	X	X/-	
SUPPRESS-CHANGE-EVENT	X	-/-	
SUPPRESS-CLICK-EVENT	X	X/X	
SUPPRESS-DBL-CLICK-EVENT	X	X/X	
SUPPRESS-ENTER-EVENT	X	X/X	
SUPPRESS-ENTER-CELL-EVENT	X	X/X	
SUPPRESS-FILL-EVENT	X	X/X	
SUPPRESS-LEAVE-EVENT	X	X/X	
SUPPRESS-LEAVE-CELL-EVENT	X	X/X	
TYPE	X	X/-	
VISIBLE	X	-/-	

## COLUMN-TYPEs and their Attributes

The attributes that are valid for column specification controls in general are not always available depending on the value of the attribute COLUMN-TYPE. Such a value might be, for example, "toggle-button control". The column then consists of cells that can be used like toggle-button controls. For these toggle-button cells, only a subset of the attributes is available.

The following table specifies which attributes are *not* applicable to a COLUMN-TYPE.

<b>COLUMN-TYPE</b>	<b>Attributes NOT Available</b>
Input field control	(All available.)
Selection box control	(All available.)
Selection box item	DIL-TEXT, HELP-ID, LENGTH, MODIFIABLE, STYLE, SUPPRESS-CHANGE-EVENT, SUPPRESS-CLICK-EVENT, SUPPRESS-DBL-CLICK-EVENT, SUPPRESS-ENTER-EVENT, SUPPRESS-ENTER-CELL-EVENT, SUPPRESS-LEAVE-EVENT, SUPPRESS-LEAVE-CELL-EVENT, VISIBLE.
Toggle button control	LENGTH, STYLE, MODIFIABLE.

## Events

This dialog element does not create events.

# Context Menu

## Description

The context menu allows you to define menus which appear when the user clicks a dialog or dialog element with the right mouse button. You can retrieve the position of the click via the INQ-CLICKPOSITION action, should this be necessary.

Context menus are defined separately from the dialog elements and associated with the relevant dialog element(s) via the CONTEXT-MENU attribute. The context menu can be modified before it is displayed via the before-open event.

The context menu is constructed and handled almost identically to the submenu control. Like the submenu control, a context menu can contain menu items which open up submenus when selected. Each submenu in such a multi-level context menu receives its own before-open event each time the submenu is displayed.

## Attributes for Context Menu

Attribute Name	Query	Set/Modify	In Attr. Window
CLIENT-DATA	X	X/X	
CLIENT-HANDLE	X	X/X	
CLIENT-KEY	X	X/X	
CLIENT-VALUE	X	X/X	
ENABLED	X	X/X	X
FIRST-CHILD	X	-/-	
HELP-ID	X	X/X	X
LAST-CHILD	X	-/-	
PARENT	X	X/-	
PREDECESSOR	X	-/-	
STYLE	X	X/-	
SUCCESSOR	X	-/-	
SUPPRESS-BEFORE-OPEN-EVENT	X	X/X	
TYPE	X	X/-	

## Events

Before-open event (may be suppressed).

# Control Box Control

## Description

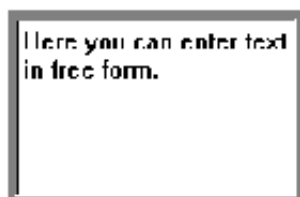
The control box control is a general-purpose container control. Any dialog elements placed within the control box become child controls of the control box, thus allowing related dialog elements to be grouped together for programming convenience or ease of user manipulation in the dialog editor. For example, when a control box is made invisible, all the dialog elements it contains become invisible, and when it is moved, these dialog elements are moved with it.

Special attention has been paid to making it possible to create multiple "pages" of dialog elements at the same position in a dialog using control boxes marked with the 'Exclusive' style. Only one such page can then be visible at any one time, both at edit-time and at run-time, and the Dialog Editor automatically changes the active page according to the current selection. This feature can be used to support wizard dialogs and ActiveX tab controls. Please refer to the article Working with Control Boxes for more information.

## Attributes for Control Box Control

Attribute Name	Query	Set/Modify	In Attr. Window
BACKGROUND-COLOUR-NAME	X	X/X	X
BACKGROUND-COLOUR-VALUE	X	X/X	X
CLIENT-DATA	X	X/X	
CLIENT-HANDLE	X	X/X	
CLIENT-KEY	X	X/X	
CLIENT-VALUE	X	X/X	
ENABLED	X	X/X	X
FIRST-CHILD	X	-/-	
FOLLOWS	X	X/X	
LAST-CHILD	X	-/-	
PARENT	X	X/-	
PREDECESSOR	X	-/-	
RECTANGLE-H	X	X/X	X
RECTANGLE-W	X	X/X	X
RECTANGLE-X	X	X/X	X
RECTANGLE-Y	X	X/X	X
STYLE	X	X/-	X
SUCCESSOR	X	-/-	
TYPE	X	X/-	
VISIBLE	X	X/X	X

## Edit Area Control



### Description

In an edit area control, the end user may type in free-form text. It may contain any number of lines, and scroll bars may be set: if there are more lines in the edit area control than can be displayed, the end user may scroll to the desired line.

To use the free form text elsewhere in your application, you will have to set the text of the edit area control into alphanumeric Natural fields line by line. You can then query these lines of text one after the other.

You can manipulate the text in the edit area control by using PROCESS GUI statement actions. This helps you insert a new line, query a certain line, set a selection, query the selected text, and so on. Note that for moving text from one edit area control to another, the STRING attribute will only be able to hold 253 characters. If you want to move text longer than 253 characters, you use the PROCESS GUI statement actions named EDIT-\*.

### Attributes for Edit Area Control

Attribute Name	Query	Set/Modify	In Attr. Window
BACKGROUND-COLOUR-NAME	X	X/X	X
BACKGROUND-COLOUR-VALUE	X	X/X	X
CLIENT-DATA	X	X/X	
CLIENT-HANDLE	X	X/X	
CLIENT-KEY	X	X/X	
CLIENT-VALUE	X	X/X	
CONTEXT-MENU	X	X/X	X
DIL-TEXT	X	X/X	X
ENABLED	X	X/X	X
FIRST-CHILD	X	-/-	
FOLLOWS	X	X/X	
FONT-HANDLE	X	X/X	X
FOREGROUND-COLOUR-NAME	X	X/X	X
FOREGROUND-COLOUR-VALUE	X	X/X	X
HELP-ID	X	X/X	X
HORIZ-SCROLLABLE	X	X/-	X
LAST-CHILD	X	-/-	
LENGTH	X	X/X	

Attribute Name	Query	Set/Modify	In Attr. Window
MODIFIABLE	X	X/X	X
MODIFIED	X	-/X	
MODIFIED-SUCCESSOR	X	-/-	
PARENT	X	X/-	
PREDECESSOR	X	-/-	
RECTANGLE-H	X	X/X	X
RECTANGLE-W	X	X/X	X
RECTANGLE-X	X	X/X	X
RECTANGLE-Y	X	X/X	X
STRING	X	X/X	X
STYLE	X	X/-	X
SUCCESSOR	X	-/-	
SUPPRESS-CHANGE-EVENT	X	-/-	
SUPPRESS-ENTER-EVENT	X	X/X	
SUPPRESS-LEAVE-EVENT	X	X/X	
TYPE	X	X/-	X
VERT-SCROLLABLE	X	X/-	X
VISIBLE	X	X/X	X

## Events

Change event (may be suppressed), enter event (may be suppressed), leave event (may be suppressed).

# Font Control

## Description

A font control is used to display the STRING attribute value of a dialog element in a certain font face, size and style. It is generated automatically if you select a font in the attributes window of a dialog element. You should, however, not rely on the automatic generation of specific font control names. You can also create a font control dynamically by using the NGU-FONT-SELECT dialog from library SYSTEM. When you assign the handle value of this font control to the FONT-HANDLE attribute of another dialog element, the STRING of the other dialog element is displayed accordingly.

### Example:

```
#TC-1.FONT-HANDLE := #FNT-1
/* The STRING of the text-constant control
/* #TC-1 will be displayed in the face,
/* style and size of #FNT-1.
```

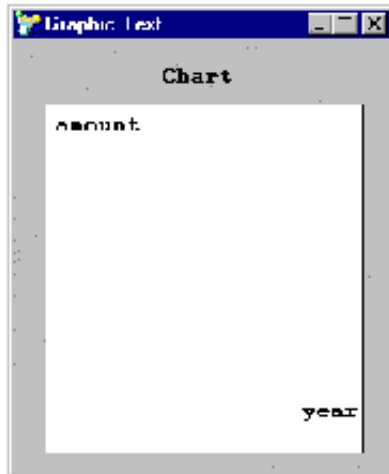
## Attributes for Font Control

Attribute Name	Query	Set/Modify	In Attr. Window
CLIENT-HANDLE	X	X/X	
CLIENT-KEY	X	X/X	
CLIENT-VALUE	X	X/X	
FIRST-CHILD	X	-/-	
LAST-CHILD	X	-/-	
PARENT	X	X/-	
PREDECESSOR	X	-/-	
STRING	X	X/-	X
SUCCESSOR	X	-/-	
TYPE	X	X/-	

## Events

This dialog element does not create events.

## Graphic-Text Control



### Description

A graphic-text control represents a one-line piece of text to be created on top of a canvas control. If it goes beyond the area of the canvas control, it is clipped.

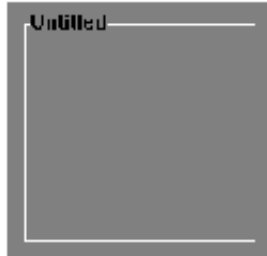
### Attributes for Graphic-Text Control

Attribute Name	Query	Set/Modify
CLIENT-DATA	X	X/X
CLIENT-HANDLE	X	X/X
CLIENT-KEY	X	X/X
CLIENT-VALUE	X	X/X
FIRST-CHILD	X	-/-
FONT-HANDLE	X	X/X
FOREGROUND-COLOUR-NAME	X	X/X
FOREGROUND-COLOUR-VALUE	X	X/X
LAST-CHILD	X	-/-
PARENT	X	X/-
PREDECESSOR	X	-/-
RECTANGLE-H	X	X/X
RECTANGLE-W	X	X/X
RECTANGLE-X	X	X/X
RECTANGLE-Y	X	X/X
STRING	X	X/X
SUCCESSOR	X	X/-
STYLE	X	-/-
TYPE	X	X/-
VISIBLE	X	X/X

## Events

This dialog element does not create events.

## Group-Frame Control



### Description

A group-frame control is used to optically group related dialog elements within a dialog. It can, for example, be used to frame a group of radio-button controls that have a common GROUP-ID, but where it is not optically evident to the end user that these radio-button controls are related.

You may use the group-frame control without the text in the upper left corner. Then the group-frame control acts as a simple frame. If you use it with text, this text may contain a mnemonic key (&). If the mnemonic key is pressed, the following dialog element in the navigation sequence gets the focus.

### Attributes for Group-Frame Control

Attribute Name	Query	Set/Modify	In Attr. Window
BACKGROUND-COLOUR-NAME	X	X/X	X
BACKGROUND-COLOUR-VALUE	X	X/X	X
CLIENT-HANDLE	X	X/X	
CLIENT-KEY	X	X/X	
CLIENT-VALUE	X	X/X	
ENABLED	X	X/X	X
FIRST-CHILD	X	-/-	
FOLLOWS	X	X/X	
FONT-HANDLE	X	X/X	X
FOREGROUND-COLOUR-NAME	X	X/X	X
FOREGROUND-COLOUR-VALUE	X	X/X	X
LAST-CHILD	X	-/-	
PARENT	X	X/-	
PREDECESSOR	X	-/-	
RECTANGLE-H	X	X/X	X
RECTANGLE-W	X	X/X	X
RECTANGLE-X	X	X/X	X
RECTANGLE-Y	X	X/X	X
STRING	X	X/X	X
SUCCESSOR	X	-/-	
TYPE	X	X/-	
VISIBLE	X	X/X	X

## Events

This dialog element does not create events.

# GUI Control

## Description

In event handler code, you can use the `HANDLE OF GUI` variable to refer to the handle of any type of dialog element. This can be useful, for example, if you are querying an attribute value in all dialog elements on one level: you go through the dialog elements one after the other; in the course of this query, it is not clear which type of dialog element is going to be queried next. Then a GUI handle allows you to query the next dialog element regardless of its type. This saves a lot of coding, because otherwise, you would have to query each dialog element's attribute value separately.

The syntax checker will accept all existing attributes for a `HANDLE OF GUI` variable because all existing dialog elements must be covered. Nevertheless, if you query or modify an attribute which is not valid for the dialog element to which the `HANDLE OF GUI` variable refers, a runtime error will occur. It is therefore only advisable to use a `HANDLE OF GUI` variable if the attribute you are querying or modifying applies to most, if not all of the dialog elements.

### Example:

```
... 1 #CONTROL
HANDLE OF GUI
...
#CONTROL := #DLG$WINDOW.FIRST-CHILD REPEAT UNTIL #CONTROL = NULL-HANDLE
...
#CONTROL := #CONTROL.SUCCESSOR
END-REPEAT
```

## Attributes for GUI Control

Attribute Name	Query	Set/Modify
ACCELERATOR	X	X/X
ACTIVE-CHILD	X	-/X
AUTOSELECT	X	X/X
BACKGROUND-COLOUR-NAME	X	X/X
BACKGROUND-COLOUR-VALUE	X	X/X
BAR-ID	X	X/-
BITMAP-FILE-NAME	X	X/X
CELL-ATTRIBUTES	X	X/X
CHECKED	X	X/X
CLIENT-DATA	X	X/X
CLIENT-HANDLE	X	X/X
CLIENT-KEY	X	X/X
CLIENT-VALUE	X	X/X
COLUMN	X	X/X
COLUMN-COUNT	X	-/-
COLUMN-TYPE	X	X/-
CONTEXT-MENU	X	X/X
DEFAULT-BUTTON	X	X/X
DIL-TEXT	X	X/X
DOCKING	X	X/X
DRAGGABLE	X	X/X
EDIT-MASK	X	X/X
EMBEDDED-OBJECT	X	X/X
ENABLED	X	X/X
FIRST-CHILD	X	-/-
FIRST-COLUMN-WIDTH	X	X/X
FIRST-VISIBLE-COLUMN	X	X/X
FIRST-VISIBLE-ROW	X	X/X
FOLLOWS	X	X/X
FONT-HANDLE	X	X/X
FONT-STRING	X	X/-
FOREGROUND-COLOUR-NAME	X	X/X
FOREGROUND-COLOUR-VALUE	X	X/X
FROZEN-COLUMNS	X	X/X

Attribute Name	Query	Set/Modify
GROUP-ID	X	X/X
HAS-DIL	X	X/X
HAS-FIRST-COLUMN	X	X/X
HAS-STATUS-BAR	X	X/X
HAS-SYSTEM-BUTTON	X	X/-
HAS-TOOLBAR	X	X/X
HAS-TOOLTIP	X	X/X
HEADER-FONT-HANDLE	X	X/X
HEADER-HEIGHT	X	X/X
HELP-ID	X	X/X
HORIZ-SCROLLABLE	X	X/-
ICONIZED	X	X/X
ITEM-H	X	X/X
ITEM-W	X	X/X
LAST-CHILD	X	-/-
LENGTH	X	X/X
LINE	X	X/X
LINKED	X	X/-
LOCATION	X	X/X
MARGIN-X	X	X/X
MARGIN-Y	X	X/X
MAX	X	X/X
MAXIMIZABLE	X	X/-
MAXIMIZED	X	X/X
MENU-HANDLE	X	X/X
MENU-ITEM-OLE	X	X/X
MENU-ITEM-TYPE	X	X/-
MIN	X	X/X
MINIMIZABLE	X	X/-
MINIMIZED	X	X/X
MODIFIABLE	X	X/-
MODIFIED	X	-/X
MODIFIED-SUCCESSOR	X	-/-
MULTI-SELECTION	X	X/-
NAME	X	-/-

Attribute Name	Query	Set/Modify
OBJECT-SIZE	X	X/X
OFFSET-X	X	X/X
OFFSET-Y	X	X/X
P1-X	X	X/X
P1-Y	X	X/X
P2-X	X	X/X
P2-Y	X	X/X
PAGE	X	X/X
PARENT	X	X/-
PREDECESSOR	X	-/-
RECTANGLE-H	X	X/X
RECTANGLE-W	X	X/X
RECTANGLE-X	X	X/X
RECTANGLE-Y	X	X/X
ROW	X	X/X
ROW-COUNT	X	X/-
ROW-HEIGHT	X	X/X
SAME-AS	X	X/-
SCROLLRANGE-X	X	X/X
SCROLLRANGE-Y	X	X/X
SELECTED	X	X/X
SELECTED-SUCCESSOR	X	-/-
SERVER-OBJECT	X	X/X
SERVER-PROGID	X	X/X
SHARED	X	X/X
SIZE-MODIFIABLE	X	X/-
SLIDER	X	X/X
SORTED	X	X/-
STATUS-HANDLE	X	X/X
STATUS-TEXT	-	-/X
STRING	X	X/X
STYLE	X	X/-
SUCCESSOR	X	X/-
SUPPRESS-ACTIVATE-EVENT	X	X/X
SUPPRESS-BEFORE-OPEN-EVENT	X	X/X

Attribute Name	Query	Set/Modify
SUPPRESS-CHANGE-EVENT	X	-/-
SUPPRESS-CLICK-EVENT	X	X/X
SUPPRESS-CLOSE-EVENT	X	X/X
SUPPRESS-COMMAND-STATUS-EVENT	X	X/X
SUPPRESS-DBL-CLICK-EVENT	X	X/X
SUPPRESS-DELETE-ROW-EVENT	X	X/X
SUPPRESS-ENTER-EVENT	X	X/X
SUPPRESS-ENTER-CELL-EVENT	X	X/X
SUPPRESS-FILL-EVENT	X	X/X
SUPPRESS-IDLE-EVENT	X	X/X
SUPPRESS-INSERT-ROW-EVENT	X	X/X
SUPPRESS-LEAVE-EVENT	X	X/X
SUPPRESS-LEAVE-CELL-EVENT	X	X/X
SUPPRESS-SIZE-EVENT	X	X/X
SUPPRESS-TOP-EVENT	X	X/X
TIMER-INTERVAL	X	X/X
TOOLBAR-HANDLE	X	X/X
TOOLBAR-POS	X	X/-
TOOLTIP	X	X/X
TYPE	X	X/-
VARIABLE	-	X/X
VERSION	X	-/-
VERT-SCROLLABLE	X	X/-
VISIBLE	X	X/X
ZOOM-FACTOR	X	X/X

# Input-Field Control



## Description

An input-field control is used to enter data in a single line. It corresponds to a field in a Natural program's INPUT statement. You can map database fields or other program variables to an input-field control by means of a linked variable. The input is automatically copied to a linked variable if you select the "Linked Variable" option in the attribute window's "Source" dialog box and you enter the name of the linked variable, for example, the database field. When the end user has finished entering data and the input-field control loses the focus, the data entered are validated.

You can also validate the input data by assigning an EDIT-MASK attribute to a (linked variable) input-field control: the check is performed when the input-field control loses the focus.

When the linked variables have been modified by code and you want to display the new values, you use the PROCESS GUI statement action REFRESH-LINKS.

Whenever input is rejected, for example, because the linked variable has another Natural data type, or because the EDIT-MASK was not matched, a message box is displayed that prompts the end user to "Retry" or "Cancel". The end user must press "Retry" to keep the current content of the input-field control and to modify it. The end user must press "Cancel" to reset the input-field control to the last valid content.

### Notes:

When you create an input-field control, and you assign it a STYLE value of "Center" ('c') or "Right" ('r'), the input-field control's height must be greater than the height of the system font. Otherwise, the STRING will not be displayed.

Input in an input-field control is limited to 253 characters.

## Attributes for Input-Field Control

Attribute Name	Query	Set/Modify	In Attr. Window
BACKGROUND-COLOUR-NAME	X	X/X	X
BACKGROUND-COLOUR-VALUE	X	X/X	X
CLIENT-DATA	X	X/X	
CLIENT-HANDLE	X	X/X	
CLIENT-KEY	X	X/X	
CLIENT-VALUE	X	X/X	
CONTEXT-MENU	X	X/X	X
DIL-TEXT	X	X/X	X
EDIT-MASK	X	X/X	X
ENABLED	X	X/X	X
FIRST-CHILD	X	-/-	
FOLLOWS	X	X/X	
FONT-HANDLE	X	X/X	X

Attribute Name	Query	Set/Modify	In Attr. Window
FOREGROUND-COLOUR-NAME	X	X/X	X
FOREGROUND-COLOUR-VALUE	X	X/X	X
HELP-ID	X	X/X	X
LAST-CHILD	X	-/-	
LENGTH	X	X/X	X
LINKED	X	X/-	X
MODIFIABLE	X	X/X	X
MODIFIED	X	-/X	
MODIFIED-SUCCESSOR	X	-/-	
PARENT	X	X/-	
PREDECESSOR	X	-/-	
RECTANGLE-H	X	X/X	X
RECTANGLE-W	X	X/X	X
RECTANGLE-X	X	X/X	X
RECTANGLE-Y	X	X/X	X
STRING	X	X/X	X
STYLE	X	X/-	X
SUCCESSOR	X	-/-	
SUPPRESS-CHANGE-EVENT	X	-/-	
SUPPRESS-ENTER-EVENT	X	X/X	
SUPPRESS-LEAVE-EVENT	X	X/X	
TYPE	X	X/-	
VARIABLE	-	X/X	X
VISIBLE	X	X/X	X

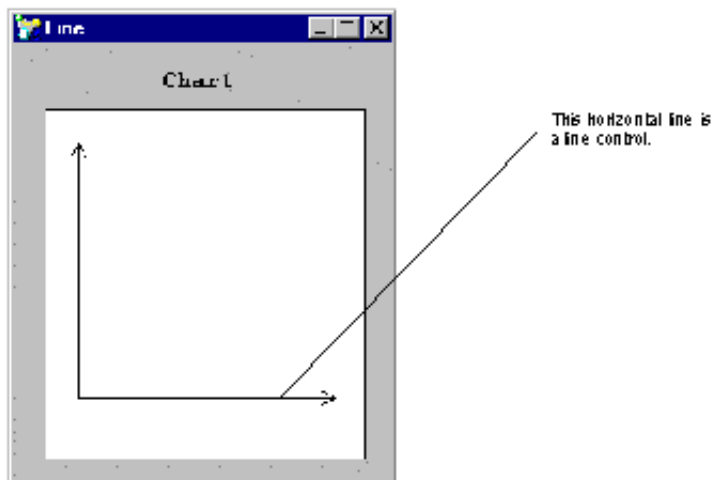
## Events

Change event (may be suppressed), enter event (may be suppressed), leave event (may be suppressed).

### Note:

The change event occurs for an input-field control if either the end user or the code changes the content. It does not occur on the initial setting of the input-field control. It is not recommended to manipulate the system focus from within the change event handler. Furthermore, the content of a linked variable is not updated in a change event.

## Line Control



### Description

A line control represents a line to be created in a canvas control. If it goes beyond the area of the canvas control, it is clipped.

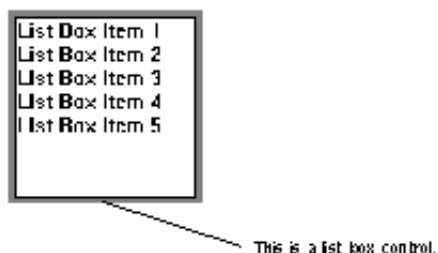
### Attributes for Line Control

Attribute Name	Query	Set/Modify
CLIENT-DATA	X	X/X
CLIENT-HANDLE	X	X/X
CLIENT-KEY	X	X/X
CLIENT-VALUE	X	X/X
FIRST-CHILD	X	-/-
FOREGROUND-COLOUR-NAME	X	X/X
FOREGROUND-COLOUR-VALUE	X	X/X
LAST-CHILD	X	-/-
PARENT	X	X/-
PREDECESSOR	X	-/-
P1-X	X	X/X
P1-Y	X	X/X
P2-X	X	X/X
P2-Y	X	X/X
STYLE	X	-/-
SUCCESSOR	X	X/-
TYPE	X	X/-
VISIBLE	X	X/X

## Events

This dialog element does not create events.

## List-Box Control



### Description

A list-box control contains a number of list-box items from which the end user can select one or more. You can insert items into a list-box control by:

- defining them in the dialog editor at design time, and by
- creating them dynamically at runtime using the PROCESS GUI statement actions ADD, ADD-ITEMS and ADD-ITEMS-EX.

The fill event enables you to implement dynamically growing list-box controls.

The MULTI-SELECTION attribute specifies for a list-box control whether the end user may select one or several items at a time.

#### Note:

A list-box control must have MULTI-SELECTION = TRUE to be able to use SELECTED-SUCCESSOR with the children list-box items.

To sort list-box item STRINGS alphabetically, you use the SORTED attribute for the list-box control. When the STRING of a list-box item is changed, the items are automatically sorted again. You can then go through the list of items with the SUCCESSOR attribute and the new sort sequence is reflected.

When a click event or a double-click event occurs for a list-box item, the list-box control receives the event and the list-box control's event handler can determine the clicked item by querying the attribute SELECTED-SUCCESSOR for the list-box control. When several list-box items are selected at the same time, the second selected item can be determined by querying the attribute SELECTED-SUCCESSOR of the first item, and so on.

If you specify event handlers for the click and the double-click events, the click-event handler will be executed prior to the double-click-event handler.

This may imply that the click-event handler suppresses the execution of the double-click-event handler (SUPPRESS-DBL-CLICK-EVENT attribute).

### Attributes for List-Box Control

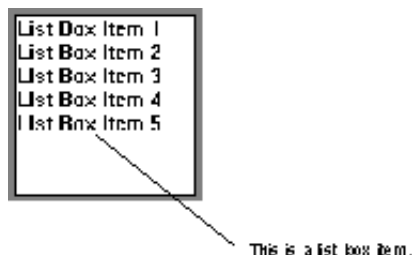
Attribute Name	Query	Set/Modify	In Attr. Window
AUTOSELECT	X	X/X	
ACCELERATOR	X	X/X	X
BACKGROUND-COLOUR-NAME	X	X/X	X
BACKGROUND-COLOUR-VALUE	X	X/X	X
CLIENT-DATA	X	X/X	

Attribute Name	Query	Set/Modify	In Attr. Window
CLIENT-HANDLE	X	X/X	
CLIENT-KEY	X	X/X	
CLIENT-VALUE	X	X/X	
CONTEXT-MENU	X	X/X	X
DIL-TEXT	X	X/X	X
ENABLED	X	X/X	X
FIRST-CHILD	X	-/-	
FOLLOWS	X	X/X	
FONT-HANDLE	X	X/X	X
FOREGROUND-COLOUR-NAME	X	X/X	X
FOREGROUND-COLOUR-VALUE	X	X/X	X
HELP-ID	X	X/X	X
LAST-CHILD	X	-/-	
MODIFIED	X	-/X	
MODIFIED-SUCCESSOR	X	-/-	
MULTI-SELECTION	X	X/-	X
PARENT	X	X/-	
PREDECESSOR	X	-/-	
RECTANGLE-H	X	X/X	X
RECTANGLE-W	X	X/X	X
RECTANGLE-X	X	X/X	X
RECTANGLE-Y	X	X/X	X
SELECTED-SUCCESSOR	X	-/-	
SORTED	X	X/-	X
STYLE	X	-/-	X
SUCCESSOR	X	-/-	
SUPPRESS-CLICK-EVENT	X	X/X	
SUPPRESS-DBL-CLICK-EVENT	X	X/X	
SUPPRESS-FILL-EVENT	X	X/X	
TYPE	X	X/-	
VISIBLE	X	X/X	X

## Events

Click event (may be suppressed), double-click event (may be suppressed), fill event (may be suppressed).

## List-Box Item



### Description

A list-box item is an item inside a list-box control.

The list-box items are added to the list-box control either by specifying them in the dialog editor's List Box Attributes window, or by using the PROCESS GUI statement actions ADD-ITEMS, ADD-ITEMS-EX or ADD.

Items created using the dialog editor are represented in a Natural variable. The variable name is generated as: *list-box-handle-name-ITEMS (1:number-of-list-box-items)*. Example: #LB-1-ITEMS (1:5). The fifth item would then be called: #LB-1-ITEMS (5).

Items created with the PROCESS GUI statement can be given any variable name in the HANDLE definitions. Example: #MYITEM-1 HANDLE OF LISTBOXITEM.

To sort list-box item STRINGS alphabetically, you use the SORTED attribute for the list-box control. When the STRING of a list-box item is changed, the items are automatically sorted again. You can then go through the list of items with the SUCCESSOR attribute and the new sort sequence is reflected. For more information on how to go through a list of items, see **Working with List Box Controls and Selection-Box Controls** in the section Event-driven Programming Techniques of your Natural User's Guide.

A list-box item does not receive any events. Instead, the list-box control receives them, triggering event handlers there.

#### Note:

To select or deselect individual list-box items at runtime, hold down CTRL while clicking with the left mouse button.

### Attributes for List-Box Item

Attribute Name	Query	Set/Modify	In Attr. Window
CLIENT-DATA	X	X/X	
CLIENT-HANDLE	X	X/X	
CLIENT-KEY	X	X/X	
CLIENT-VALUE	X	X/X	
FIRST-CHILD	X	-/-	
LAST-CHILD	X	-/-	
PARENT	X	X/-	
PREDECESSOR	X	-/-	
SELECTED	X	X/X	
SELECTED-SUCCESSOR	X	-/-	
STRING	X	X/X	X
SUCCESSOR	X	X/-	
TYPE	X	X/-	

**Note:**

To be able to use SELECTED-SUCCESSOR, the parent list-box control must have MULTI-SELECTION = TRUE.

**Events**

This dialog element does not create events.

## Menu Bar



This is a menu bar control without any menu items or submenu controls.

### Description

A menu bar is displayed at the top of the dialog window. It is the top level of a menu structure and contains menu items (second level). A menu item may be of type submenu. Then it is a submenu control which may be pulled down; it contains menu items (third level). The number of levels is unlimited.

A menu bar only becomes visible in a dialog when the attribute MENU-HANDLE is set to a value for the dialog. The PARENT of a menu bar may be a dialog or a NULL-HANDLE. If a dialog is the PARENT, the menu structure is specific to the dialog and is deleted when the dialog is deleted. If a NULL-HANDLE is the PARENT, the menu structure is free and will be closed when the application is closed. A free menu structure can be shared among several dialogs when the handles are defined in a global data area rather than in a local data area.

For an MDI child window, the menu bar is displayed only at the top of the MDI frame window, not in the MDI child window. The MDI children all share one menu bar, which is displayed in the MDI frame window. Every time another MDI child window is activated, the menu bar changes to reflect the menu bar defined for the particular MDI child window.

#### Note:

You may create only one menu bar per dialog. By default, the dialog editor generates a menu bar named "#DLG-MENU-BAR", the parent being the dialog window.

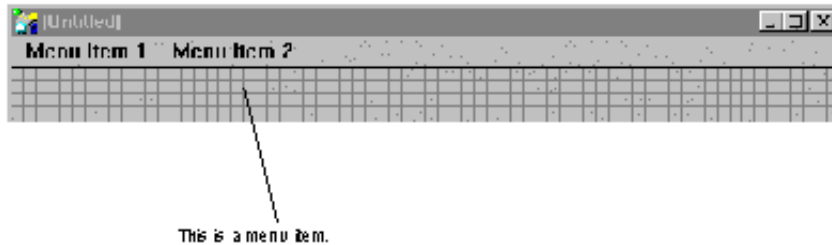
### Attributes for Menu Bar

Attribute Name	Query	Set/Modify	In Attr. Window
CLIENT-DATA	X	X/X	
CLIENT-HANDLE	X	X/X	
CLIENT-KEY	X	X/X	
CLIENT-VALUE	X	X/X	
FIRST-CHILD	X	-/-	
HELP-ID	X	X/X	X
LAST-CHILD	X	-/-	
PARENT	X	X/-	
PREDECESSOR	X	-/-	
SUCCESSOR	X	-/-	
TYPE	X	X/-	
VISIBLE	X	X/X	

## Events

This dialog element does not create events.

# Menu Item



## Description

A menu item is an item inside a menu bar or a submenu control. It is a child of a menu bar or a submenu control.

There are several types of menu items (values of the attribute MENU-ITEM-TYPE). Menu items of type MT-NORMAL trigger the click-event handler when an end user clicks on them. The MT-SUBMENU type of menu item is associated with a submenu control, which is pulled down when an end user clicks on the menu item.

It is not recommended to define menu items of type MT-NORMAL on the first level of a menu bar. Instead, you should define menu items of type MT-SUBMENU. When you give a name to a menu item by setting the STRING attribute to a name used by the windowing system, such as "Cascade", you must not modify this STRING value dynamically.

## Attributes for Menu Item

Attribute Name	Query	Set/Modify	In Attr. Window
ACCELERATOR <sup>3)</sup>	X	X/X	X
BACKGROUND-COLOUR-NAME	X	X/X	X
BACKGROUND-COLOUR-VALUE	X	X/X	X
BITMAP-FILE-NAME	X	X/X	X
CHECKED <sup>1)</sup>	X	X/X	X
CLIENT-DATA <sup>2)</sup>	X	X/X	
CLIENT-HANDLE	X	X/X	
CLIENT-KEY	X	X/X	
CLIENT-VALUE	X	X/X	
DIL-TEXT <sup>2)</sup>	X	X/X	X
ENABLED <sup>3)</sup>	X	X/X	X
FIRST-CHILD	X	-/-	
LAST-CHILD	X	-/-	
MENU-HANDLE <sup>4)</sup>	X	X/-	
MENU-ITEM-OLE	X	X/X	
MENU-ITEM-TYPE <sup>5)</sup>	X	X/-	X
PARENT	X	X/-	
PREDECESSOR	X	-/-	
SAME-AS	X	X/X	X
SHARED	X	X/X	X
STRING <sup>3)</sup>	X	X/X	X
STYLE	X	X/-	X
SUCCESSOR	X	-/-	
TOOLTIP	X	-/-	
TYPE	X	X/-	

1) For MENU-ITEM-TYPE = MT-NORMAL only. Not for menu items inside the menu bar.

2) For MENU-ITEM-TYPEs MT-NORMAL and MT-MDI only.

3) Not for MENU-ITEM-TYPE = MT-SEPARATOR.

4) For MENU-ITEM-TYPEs MT-SUBMENU and MT-WINDOWMENU only.

5) If MENU-ITEM-TYPE = MT-WINDOWMENU, the menu item must be inside the menu bar (at the top level).

## Events

Click event.

## OLE Container Control

1' Reg on 4
25 Reg on D
77 Reg on C
113 Reg onns Total

This is an OLE container control with an embedded spreadsheet

### Description

An OLE container control enables you to integrate OLE objects in a Natural dialog.

### Attributes for OLE Container Control

Attribute Name	Query	Set/Modify	In Attr. Window
ACCELERATOR	X	X/X	X
BACKGROUND-COLOUR-NAME	X	X/X	X
BACKGROUND-COLOUR-VALUE	X	X/X	X
CLIENT-DATA	X	X/X	
CLIENT-HANDLE	X	X/X	
CLIENT-KEY	X	X/X	
CLIENT-VALUE	X	X/X	
DIL-TEXT	X	X/X	X
EMBEDDED-OBJECT	X	X/X	X
ENABLED	X	X/X	X
FIRST-CHILD	X	-/-	
FOLLOWS	X	X/X	
HELP-ID	X	X/X	X
ICONIZED	X	X/-	
INPLACE-ACTIVE	X	-/-	
LAST-CHILD	X	-/-	
MODIFIABLE	X	X/X	X
OBJECT-SIZE	X	-/-	
PARENT	X	X/-	
PREDECESSOR	X	-/-	
RECTANGLE-H	X	X/X	X
RECTANGLE-W	X	X/X	X
RECTANGLE-X	X	X/X	X
RECTANGLE-Y	X	X/X	X
SERVER-OBJECT	X	X/X	X

Attribute Name	Query	Set/Modify	In Attr. Window
SERVER-PROGID	X	X/X	X
STYLE	X	X/-	X
SUCCESSOR	X	-/-	
SUPPRESS-ACTIVATE-EVENT	X	X/X	
SUPPRESS-CHANGE-EVENT	X	-/-	
SUPPRESS-CLICK-EVENT	X	X/X	
SUPPRESS-CLOSE-EVENT	X	X/X	
SUPPRESS-DBL-CLICK-EVENT	X	X/X	
TYPE	X	X/-	
VISIBLE	X	X/X	X
ZOOM-FACTOR	X	X/X	X

## Events

Activate event (may be suppressed), change event (may be suppressed), click event (may be suppressed), close event (may be suppressed), double-click event (may be suppressed).

## Push-Button Control



### Description

A push-button control is a representation of a button. The end user can click on it to trigger a certain action (specified in the click-event handler code). You can use a push-button control's handle value in the DEFAULT-BUTTON attribute of a dialog; your push-button control then serves as the dialog's default button.

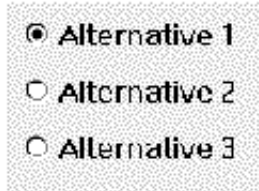
### Attributes for Push-Button Control

Attribute Name	Query	Set/Modify	In Attr. Window
ACCELERATOR	X	X/X	X
BACKGROUND-COLOUR-NAME	X	X/X	X
BACKGROUND-COLOUR-VALUE	X	X/X	X
CLIENT-DATA	X	X/X	
CLIENT-HANDLE	X	X/X	
CLIENT-KEY	X	X/X	
CLIENT-VALUE	X	X/X	
CONTEXT-MENU	X	X/X	X
DIL-TEXT	X	X/X	X
ENABLED	X	X/X	X
FIRST-CHILD	X	-/-	
FOLLOWS	X	X/X	
FONT-HANDLE	X	X/X	X
FOREGROUND-COLOUR-NAME	X	X/X	X
FOREGROUND-COLOUR-VALUE	X	X/X	X
HELP-ID	X	X/X	X
LAST-CHILD	X	-/-	
PARENT	X	X/-	
PREDECESSOR	X	-/-	
RECTANGLE-H	X	X/X	X
RECTANGLE-W	X	X/X	X
RECTANGLE-X	X	X/X	X
RECTANGLE-Y	X	X/X	X
STRING	X	X/X	X
STYLE	X	X/-	X
SUCCESSOR	X	-/-	
TYPE	X	X/-	
VISIBLE	X	X/X	X

## Events

Click event.

## Radio-Button Control



These three radio button controls have been created independently; they share a GROUP-ID, and one is checked.

### Description

A radio-button control is a selection item. You can logically group several radio-button controls by assigning them a common GROUP-ID attribute. Out of this group, none or one may be selected by default. To pre-select a radio-button control, you assign it a CHECKED attribute with a value of CHECKED. Please note that only the radio-button control CHECKED last will remain CHECKED; the one CHECKED before will be unchecked automatically.

You use it, for example, to let the end user switch an option on or off, affecting the status of other radio-button controls of the same group. A change in the state of a toggle-button control, by contrast, will not affect the state of other toggle-button controls in the same dialog.

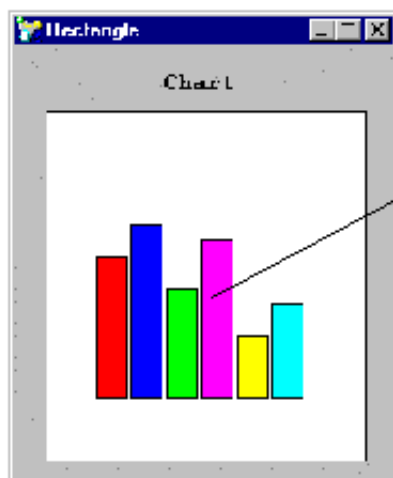
### Attributes for Radio-Button Control

Attribute Name	Query	Set/Modify	In Attr. Window
ACCELERATOR	X	X/X	X
BACKGROUND-COLOUR-NAME	X	X/X	X
BACKGROUND-COLOUR-VALUE	X	X/X	X
CHECKED	X	X/X	X
CLIENT-DATA	X	X/X	
CLIENT-HANDLE	X	X/X	
CLIENT-KEY	X	X/X	
CLIENT-VALUE	X	X/X	
CONTEXT-MENU	X	X/X	X
DIL-TEXT	X	X/X	X
ENABLED	X	X/X	X
FIRST-CHILD	X	-/-	
FOLLOWS	X	X/X	
FONT-HANDLE	X	X/X	X
FOREGROUND-COLOUR-NAME	X	X/X	X
FOREGROUND-COLOUR-VALUE	X	X/X	X
GROUP-ID	X	X/-	X
HELP-ID	X	X/X	X
LAST-CHILD	X	-/-	
MODIFIED	X	-/X	
MODIFIED-SUCCESSOR	X	-/-	
PARENT	X	X/-	
PREDECESSOR	X	-/-	
RECTANGLE-H	X	X/X	X
RECTANGLE-W	X	X/X	X
RECTANGLE-X	X	X/X	X
RECTANGLE-Y	X	X/X	X
STRING	X	X/X	X
SUCCESSOR	X	-/-	
SUPPRESS-CLICK-EVENT	X	X/X	
TYPE	X	X/-	
VISIBLE	X	X/X	X

## Events

Click event (may be suppressed).

## Rectangle Control



This is a rectangle control.

### Description

A rectangle control represents a filled rectangle to be created in of a canvas control. If it goes beyond the area of the canvas control, it is clipped.

### Attributes for Rectangle Control

Attribute Name	Query	Set/Modify
BACKGROUND-COLOUR-NAME	X	X/X
BACKGROUND-COLOUR-VALUE	X	X/X
CLIENT-DATA	X	X/X
CLIENT-HANDLE	X	X/X
CLIENT-KEY	X	X/X
CLIENT-VALUE	X	X/X
FIRST-CHILD	X	-/-
FOREGROUND-COLOUR-NAME	X	X/X
FOREGROUND-COLOUR-VALUE	X	X/X
LAST-CHILD	X	-/-
PREDECESSOR	X	-/-
PARENT	X	X/-
RECTANGLE-H	X	X/X
RECTANGLE-W	X	X/X
RECTANGLE-X	X	X/X
RECTANGLE-Y	X	X/X
STYLE	X	X/-
SUCCESSOR	X	X/-
TYPE	X	X/-
VISIBLE	X	X/X

## Events

This dialog element does not create events.

## Scrollbar Control



### Description

A scroll bar control enables the end user to select a position on a scale. If, for example, you want the end user to select a percentage value on a scale from 1 to 100, you can use a scroll bar control and tell the end user in the **STRING** attribute of a text-constant control which percentage value (integer) was selected. To limit the scale, you must always determine values for the **MIN** and **MAX** attributes of your scroll bar control. You can then use the **LINE** and **PAGE** attributes to determine step sizes for scrolling with the arrow buttons (**LINE**) or the scroll bar shaft (**PAGE**). You can also set and query the position of the slider with the **SLIDER** attribute.

**Note:**

The range of the dialog scroll bar control attributes **min**, **max**, **page**, **line** and **slider** has been increased. Now for all of these attributes a range of -1073741823 to 1073741823 is permitted.

### Attributes for Scrollbar Control

Attribute Name	Query	Set/Modify	In Attr. Window
BACKGROUND-COLOUR-NAME	X	X/X	X
BACKGROUND-COLOUR-VALUE	X	X/X	X
CLIENT-DATA	X	X/X	
CLIENT-HANDLE	X	X/X	
CLIENT-KEY	X	X/X	
CLIENT-VALUE	X	X/X	
CONTEXT-MENU	X	X/X	X
DIL-TEXT	X	X/X	X
ENABLED	X	X/X	X
FIRST-CHILD	X	-/-	
FOLLOWS	X	X/X	
HELP-ID	X	X/X	X
LAST-CHILD	X	-/-	
LINE	X	X/X	X
MAX	X	X/X	X
MIN	X	X/X	X
MODIFIED	X	-/X	
MODIFIED-SUCCESSOR	X	-/-	
PAGE	X	X/X	X
PARENT	X	X/-	
PREDECESSOR	X	-/-	
RECTANGLE-H	X	X/X	X
RECTANGLE-W	X	X/X	X
RECTANGLE-X	X	X/X	X
RECTANGLE-Y	X	X/X	X
SLIDER	X	X/X	X
STYLE	-	X/X	X
SUCCESSOR	X	-/-	
SUPPRESS-CHANGE-EVENT	X	-/-	
TYPE	X	X/-	
VISIBLE	X	X/X	X

## Events

Change event (may be suppressed).

## Selection-Box Control



This is a selection box control in its original state; when you click on the arrow push button, a list box drops down.

### Description

A selection-box control is a combination of an input-field, a list-box, and a push button. When no selection has been made, it consists of an input-field with a push button next to it. With the push button, the end user may open a list-box from the input-field. This list-box contains any number of items from which the end user can select one. The selected item is then copied into the input-field.

It is possible to get a selection-box control without a push button where the list-box is always dropped down: the STYLE attribute value must have the value "X".

You can map database fields or other program variables to the input section of a selection-box control by means of a linked variable. The input is automatically copied to a linked variable if you select the "Linked Variable" option in the attribute window's "Source" dialog box and you enter the name of the linked variable, for example the database field. When the end user has finished entering data and the selection-box control loses the focus, the entered data are validated.

You can also validate the input data by assigning an EDIT-MASK attribute to the selection-box control: the check is performed when the input-field in the selection-box control loses the focus.

When the linked variables have been modified by code and you want to display the new values, you use the PROCESS GUI statement action REFRESH-LINKS.

Whenever input is rejected, for example, because the linked variable has another Natural data type, or because the EDIT-MASK was not matched, a message box is displayed that prompts the end user to "Retry" or "Cancel". The end user must press "Retry" to keep the current content of the input-field control and to modify it. The end user must press "Cancel" to reset the input-field control to the last valid content.

#### Note:

Input in a selection-box control's input-field is limited to 253 characters.

If you set the STYLE attribute to the value "X" (dropped down), the MODIFIABLE attribute value is automatically set to TRUE. If you set the MODIFIABLE attribute to FALSE, the STYLE attribute automatically no longer has the value "X".

### Attributes for Selection-Box Control

Attribute Name	Query	Set/Modify	In Attr. Window
BACKGROUND-COLOUR-NAME	X	X/X	X
BACKGROUND-COLOUR-VALUE	X	X/X	X
CLIENT-DATA	X	X/X	
CLIENT-HANDLE	X	X/X	
CLIENT-KEY	X	X/X	
CLIENT-VALUE	X	X/X	
CONTEXT-MENU	X	X/X	X
DIL-TEXT	X	X/X	X

Attribute Name	Query	Set/Modify	In Attr. Window
EDIT-MASK	X	X/-	X
ENABLED	X	X/X	X
FIRST-CHILD	X	-/-	
FOLLOWS	X	X/X	
FONT-HANDLE	X	X/X	X
FOREGROUND-COLOUR-NAME	X	X/X	X
FOREGROUND-COLOUR-VALUE	X	X/X	X
HELP-ID	X	X/X	X
LAST-CHILD	X	-/-	
LENGTH	X	X/X	X
LINKED	X	X/-	X
MODIFIABLE	X	X/-	X
MODIFIED	X	-/X	
MODIFIED-SUCCESSOR	X	-/-	
PARENT	X	X/-	
PREDECESSOR	X	-/-	
RECTANGLE-H	X	X/X	X
RECTANGLE-W	X	X/X	X
RECTANGLE-X	X	X/X	X
RECTANGLE-Y	X	X/X	X
SELECTED-SUCCESSOR	X	-/-	
SORTED	X	X/-	
STRING	X	X/X	X
STYLE	X	X/-	X
SUCCESSOR	X	-/-	
SUPPRESS-CHANGE-EVENT	X	-/-	
SUPPRESS-ENTER-EVENT	X	X/X	
SUPPRESS-LEAVE-EVENT	X	X/X	
TYPE	X	X/-	
VARIABLE		X/X	X
VISIBLE	X	X/X	X

## Events

Change event (may be suppressed), enter event (may be suppressed), leave event (may be suppressed).

**Note:**

The change event occurs for a selection-box control if the end user changes the content, if the code changes the content or if an item is selected and transferred to the input-field area. It does not occur on the initial setting of the selection-box control.

It is not recommended to manipulate the system focus from within the change event handler.

## Selection-Box Item



### Description

A selection-box item is an item inside a selection-box control.

The selection-box items are added to the list-box control either by specifying them in the dialog editor's selection-box attributes window, or by using the PROCESS GUI statement actions ADD-ITEMS, ADD-ITEMS-EX or ADD. You can find out the selected item by querying the STRING attribute of the selection-box control.

Items created using the dialog editor are represented in a Natural variable. The variable name is generated as: *selection-box-handle-name-ITEMS (1:number-of-selection-box-items)*.

Example: #SB-1-ITEMS (1:5). The fifth item would then be called: #SB-1-ITEMS (5).

Items created with the PROCESS GUI statement actions can be given any variable name in the HANDLE definitions.

Example: #MYITEM-1 HANDLE OF SELECTIONBOXITEM.

A selection-box item does not receive any events. Instead, the selection-box control receives them, triggering event handlers there.

### Attributes for Selection-Box Item

Attribute Name	Query	Set/Modify	In Attr. Window
BACKGROUND-COLOUR-NAME	X	X/X	X
BACKGROUND-COLOUR-VALUE	X	X/X	X
CLIENT-DATA	X	X/X	
CLIENT-HANDLE	X	X/X	
CLIENT-KEY	X	X/X	
CLIENT-VALUE	X	X/X	
DIL-TEXT	X	X/X	X
FIRST-CHILD	X	-/-	
FOREGROUND-COLOUR-NAME	X	X/X	X
FOREGROUND-COLOUR-VALUE	X	X/X	X
LAST-CHILD	X	-/-	
PARENT	X	X/-	
PREDECESSOR	X	-/-	
STRING	X	X/X	X
SUCCESSOR	X	X/-	
TYPE	X	X/-	

## Events

This dialog element does not create events.

# Signal

## Description

A signal is an abstract dialog element, which does not have a user interface of its own. It is used to represent an application action which can be triggered via any number of menu items and/or tool-bar items, by linking these items to the signal via their SAME-AS attribute.

By representing each program action by a signal, and linking to the signal from each menu or tool-bar item which triggers this action, the relevant attribute values only need to be specified once for the signal itself. The signal's attributes are automatically inherited by all items which are linked to it. This inheritance mechanism also applies to any modifications made to the signal after it has been created. For example, if the ENABLED attribute of the signal is set to FALSE, all menu items and tool-bar items linked to it will be automatically disabled

If a menu or tool-bar item that is linked to a signal is clicked, a Click Event is raised for the signal itself, rather than for the menu or tool-bar item, ensuring that the same code is invoked for all items which are linked to the same signal.

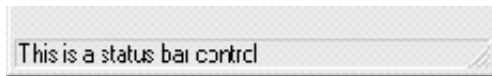
## Attributes for Signal

Attribute Name	Query	Set/Modify	In Attr. Window
ACCELERATOR	X	X/X	X
BACKGROUND-COLOUR-NAME	X	X/X	X
BACKGROUND-COLOUR-VALUE	X	X/X	X
BITMAP-FILE-NAME	X	X/X	X
CHECKED	X	X/X	X
CLIENT-DATA	X	X/X	
CLIENT-HANDLE	X	X/X	
CLIENT-KEY	X	X/X	
CLIENT-VALUE	X	X/X	X
DIL-TEXT	X	X/X	X
ENABLED	X	X/X	X
FIRST-CHILD	X	-/-	
LAST-CHILD	X	-/-	
MENU-ITEM-TYPE	X	X/X	X
PARENT	X	X/-	
PREDECESSOR	X	-/-	
SHARED	X	X/X	X
SUCCESSOR	X	-/-	
TOOLTIP	X	X/X	X
TYPE	X	X/-	
VISIBLE	X	X/X	X

## **Events**

Click event.

## Status-Bar Control



### Description

A status-bar control is an alternative to the traditional status-bar that is created by setting the dialog's HAS-STATUS-BAR attribute. However, the status-bar control offers a range of advanced features which the traditional Natural status-bars do not support. These new features include a Windows-like appearance with (optionally) recessed or raised sections - referred to as "panes" - and (optionally) a sizing grip (see illustration above). The status-bar panes are themselves dialog elements, and are therefore documented under their own section below. Note that, although it is possible to define a status-bar control with no panes, such a status-bar control can only display a single text string. In order to use any of the advanced features, it is necessary to define one or more panes.

A status-bar control can be displayed either at the top or bottom of a dialog. Furthermore, a dialog may have more than one status-bar control. Status information set via the dialog's STATUS-TEXT attribute is automatically redirected to the status-bar control (if any) specified by the dialog's STATUS-HANDLE attribute.

You can set a minimum pane height for the status-bar control using the ITEM-H attribute. The distance between the items and the status-bar control's border is determined with the MARGIN-X and MARGIN-Y attributes.

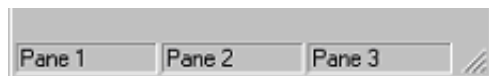
### Attributes for Status-Bar Control

Attribute Name	Query	Set/Modify	In Attr. Window
BAR-ID	X	X/-	
CLIENT-DATA	X	X/X	
CLIENT-HANDLE	X	X/X	
CLIENT-KEY	X	X/X	
CLIENT-VALUE	X	X/X	X
Context-MENU	X	X/X	X
ENABLED	X	X/X	X
FIRST-CHILD	X	-/-	
FOLLOWS	X	X/X	
FONT-HANDLE	X	X/X	X
HAS-TOOLTIP	X	X/X	X
ITEM-H	X	X/X	X
LAST-CHILD	X	-/-	
LOCATION	X	X/X	X
MARGIN-X	X	X/X	X
MARGIN-Y	X	X/X	X
PARENT	X	X/-	
PREDECESSOR	X	-/-	
RECTANGLE-H	X	-/-	
RECTANGLE-W	X	-/-	
RECTANGLE-X	X	-/-	
RECTANGLE-Y	X	-/-	
STRING	X	-/X	X
STYLE	X	X/-	X
SUCCESSOR	X	-/-	
TYPE	X	X/-	
VISIBLE	X	X/X	X

## Events

This dialog element does not create events.

## Status-Bar Pane



### Description

A status-bar pane is a logical section of a status-bar control. You can define more panes than are physically visible in the control by selectively setting the **VISIBLE** attribute of those logical panes which should not appear, according to the application context (for example, depending on which MDI child dialog is active).

A pane can be stretchy or non-stretchy. A stretchy pane is indicated by setting the pane's **RECTANGLE-W** attribute to zero. The width of a stretchy pane changes when the width of the parent dialog is changed, and is determined by the following formula:

**width = (available width - total width of all non-stretchy panes) / no. of stretchy panes**

The first (visible) stretchy pane is referred to as the message pane, and is used as the default pane in cases where no pane is explicitly specified. This is the case, for example, when text is output by setting the status-bar control's **STRING** attribute or the dialog's **STATUS-TEXT** attribute.

Both an icon and a string can be specified for a pane by setting the pane's **BITMAP-FILE-NAME** and **STRING** attributes, respectively. If the **ENABLED** attribute is set to **FALSE**, both the icon and string are either grayed out or completely suppressed (depending on a pane-style flag setting). Furthermore, the status-bar pane supports the **TOOLTIP** attributes, allowing pane-specific tool tips to be specified when the mouse pointer hovers over the pane.

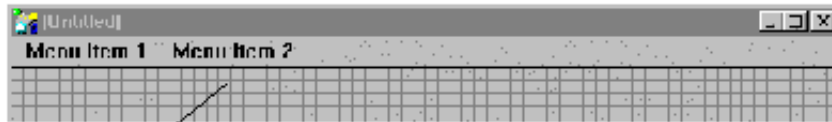
### Attributes for Status-Bar Pane

Attribute Name	Query	Set/Modify	In Attr. Window
BITMAP-FILE-NAME	X	X/X	X
CLIENT-DATA	X	X/X	
CLIENT-HANDLE	X	X/X	
CLIENT-KEY	X	X/X	
CLIENT-VALUE	X	X/X	X
ENABLED	X	X/X	X
FIRST-CHILD	X	-/-	
LAST-CHILD	X	-/-	
PARENT	X	X/-	
PREDECESSOR	X	-/-	
RECTANGLE-W	X	X/X	X
SHARED	X	X/X	X
STRING	X	-/X	X
STYLE	X	X/-	X
SUCCESSOR	X	X/-	
SUPPRESS-CLICK-EVENT	X	X/X	
SUPPRESS-DBL-CLICK-EVENT	X	X/X	
TOOLTIP	X	X/X	X
TYPE	X	X/-	
VISIBLE	X	X/X	X

## Events

Click event (may be suppressed), double-click event (may be suppressed).

## Submenu Control



This menu item represents a submenu control; that is, when you click on the menu item, a submenu will be pulled down.

### Description

A submenu control is a part of a complex menu structure. When the end user chooses a menu item of type submenu, the submenu control is pulled down, containing a vertical list of menu items.

You create a submenu control in the dialog editor by creating a menu item with the MENU-ITEM-TYPE of MT-SUBMENU or MT-WINDOWMENU. In Natural code, a submenu control is embedded into a menu structure by assigning its HANDLE to the MENU-HANDLE attribute of a menu item.

An MT-WINDOWMENU submenu control is used for MDI frame and MDI child windows. The windowing system adds the currently existing MDI child windows dynamically to such a submenu control.

The PARENT of a submenu control may be a dialog or NULL-HANDLE. If a dialog is the PARENT, the submenu control is private to the dialog and is deleted when the dialog is closed. If a NULL-HANDLE is the PARENT, the submenu control is free and will be deleted when the application is closed. A free submenu control can be shared among several menu structures in several dialogs if the handles are defined in a global data area rather than in a local data area.

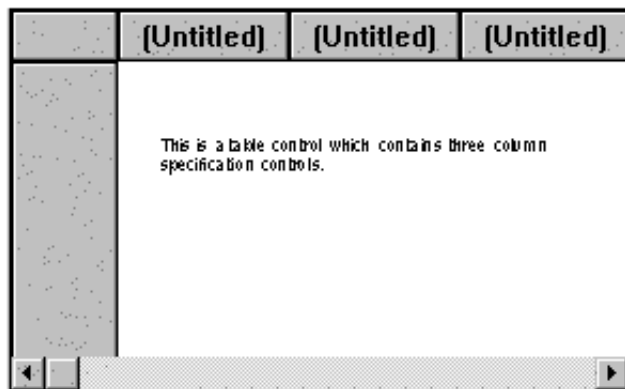
### Attributes for Submenu Control

Attribute Name	Query	Set/Modify	In Attr. Window
CLIENT-DATA	X	X/X	
CLIENT-HANDLE	X	X/X	
CLIENT-KEY	X	X/X	
CLIENT-VALUE	X	X/X	
FIRST-CHILD	X	-/-	
HELP-ID	X	X/X	
LAST-CHILD	X	-/-	
MENU-ITEM-OLE	X	X/X	
PARENT	X	X/-	
PREDECESSOR	X	-/-	
STYLE	X	X/-	
SUCCESSOR	X	-/-	
SUPPRESS-BEFORE-OPEN-EVENT	X	X/X	
TYPE	X	X/-	
VISIBLE	X	X/X	

## Events

Before-open event (may be suppressed).

## Table Control



### Description

A table control is a dialog element representing a spreadsheet. It provides an optional header row for naming the columns and an optional left column for naming the rows.

Once the table control has defined the spreadsheet as such, the columns are defined by adding column-specification controls. The cells of the table control are not defined as dialog elements but are identified by the values of the table control's COLUMN and ROW attributes. The first cell is identified by ROW=1 and COLUMN=1. The cells of each column may be used differently depending on the value of the column-specification control's COLUMN-TYPE attribute: they may be used as input-fields, selection-boxes with items or toggle-buttons.

You manipulate table controls with a set of PROCESS GUI statement actions.

### Attributes for Table Control

Attribute Name	Query	Set/Modify	In Attr. Window
BACKGROUND-COLOUR-NAME	X	X/X	X
BACKGROUND-COLOUR-VALUE	X	X/X	X
CELL-ATTRIBUTES	X	X/X	
CLIENT-DATA	X	X/X	
CLIENT-HANDLE	X	X/X	
CLIENT-KEY	X	X/X	
CLIENT-VALUE	X	X/X	
COLUMN	X	-/X	
COLUMN-COUNT	X	X/-	
CONTEXT-MENU	X	X/X	X
DIL-TEXT	X	X/X	X
ENABLED	X	X/X	X
FIRST-CHILD	X	-/-	
FIRST-COLUMN-WIDTH	X	X/X	X
FIRST-VISIBLE-COLUMN	X	X/X	X

Attribute Name	Query	Set/Modify	In Attr. Window
FIRST-VISIBLE-ROW	X	X/X	X
FOLLOWS	X	X/X	
FONT-HANDLE	X	X/X	X
FOREGROUND-COLOUR-NAME	X	X/X	X
FOREGROUND-COLOUR-VALUE	X	X/X	X
FROZEN-COLUMNS	X	X/X	X
HAS-FIRST-COLUMN	X	X/-	X
HEADER-FONT-HANDLE	X	X/X	X
HEADER-HEIGHT	X	X/X	X
HELP-ID	X	X/X	X
HORIZ-SCROLLABLE	X	X/-	X
LAST-CHILD	X	-/-	
MODIFIABLE	X	X/-	X
MODIFIED	X	-/X	
MODIFIED-SUCCESSOR	X	-/-	
PARENT	X	X/-	
PREDECESSOR	X	-/-	
RECTANGLE-H	X	X/X	X
RECTANGLE-W	X	X/X	X
RECTANGLE-X	X	X/X	X
RECTANGLE-Y	X	X/X	X
ROW	X	-/X	
ROW-COUNT	X	X/-	X
ROW-HEIGHT	X	X/X	X
STYLE	X	X/-	X
SUCCESSOR	X	X/-	
SUPPRESS-CHANGE-EVENT	X	-/-	
SUPPRESS-CLICK-EVENT	X	X/X	
SUPPRESS-DBL-CLICK-EVENT	X	X/X	
SUPPRESS-DELETE-ROW-EVENT	X	X/X	
SUPPRESS-ENTER-EVENT	X	X/X	
SUPPRESS-ENTER-CELL-EVENT	X	X/X	
SUPPRESS-FILL-EVENT	X	X/X	
SUPPRESS-INSERT-ROW-EVENT	X	X/X	
SUPPRESS-LEAVE-EVENT	X	X/X	

Attribute Name	Query	Set/Modify	In Attr. Window
SUPPRESS-LEAVE-CELL-EVENT	X	X/X	
SUPPRESS-TOP-EVENT	X	X/X	
TYPE	X	X/-	
VERT-SCROLLABLE	X	X/-	X
VISIBLE	X	X/X	X

## Attributes for Cells in a Table Control

Attribute Name	Query	Set/Modify	In Attr. Window
BACKGROUND-COLOUR-NAME	X	X/X	
BACKGROUND-COLOUR-VALUE	X	X/X	
CHECKED	X	X/X	
COLUMN	X	X/X	
DIL-TEXT	X	X/X	
FOREGROUND-COLOUR-NAME	X	X/X	
FOREGROUND-COLOUR-VALUE	X	X/X	
ROW	X	X/X	
STRING	X	X/X	

## COLUMN-TYPEs in Cells and Their Attributes

The attributes that are valid for cells in table controls in general are not always available depending on the value of the column-specification control's attribute COLUMN-TYPE. Such a value might be, for example, "input-field control". The cell can then be used like an input-field control. For this input-field cell, only a subset of the attributes is available.

The following table specifies which attributes are *not* applicable to specific COLUMN-TYPEs.

COLUMN-TYPE	Attributes NOT Available
Input field control	CHECKED
Selection box control	CHECKED
Toggle button control	STRING

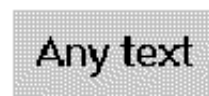
## Events

Change event, click event, delete-row event, dbl-click-event, enter event, enter-cell event, fill event, insert-row event, leave event, leave-cell event, top event. All events may be suppressed.

The following events are not always triggered:

- click event - only for toggle-button fields,
- dbl-click event - for input-fields and toggle-buttons,
- change event - only for input-fields and for selection-boxes.

## Text-Constant Control



### Description

A text-constant control displays non-modifiable text within a dialog. You use it, for example, to describe which data can be entered in an adjacent input-field control. The text `STRING` in text-constant controls may also contain a mnemonic key (`&`). If the end user presses the mnemonic key, the next input dialog element in the navigation sequence gets the focus.

### Attributes for Text-Constant Control

Attribute Name	Query	Set/Modify	In Attr. Window
BACKGROUND-COLOUR-NAME	X	X/X	X
BACKGROUND-COLOUR-VALUE	X	X/X	X
CLIENT-HANDLE	X	X/X	
CLIENT-KEY	X	X/X	
CLIENT-VALUE	X	X/X	
ENABLED	X	X/X	X
FIRST-CHILD	X	-/-	
FOLLOWS	X	X/X	
FONT-HANDLE	X	X/X	X
FOREGROUND-COLOUR-NAME	X	X/X	X
FOREGROUND-COLOUR-VALUE	X	X/X	X
LAST-CHILD	X	-/-	
PARENT	X	X/-	
PREDECESSOR	X	-/-	
RECTANGLE-H	X	X/X	X
RECTANGLE-W	X	X/X	X
RECTANGLE-X	X	X/X	X
RECTANGLE-Y	X	X/X	X
STRING	X	X/X	X
STYLE	X	X/-	X
SUCCESSOR	X	-/-	
TYPE	X	X/-	
VISIBLE	X	X/X	X

## Events

This dialog element does not create events.

# Timer

## Description

A timer is a dialog element which is invisible to the end user but which allows events to be triggered periodically in the dialog. You can, for example, update the STRING attribute of a text-constant control with the click event of a timer.

### Example:

```
#TC-1.STRING:= *TIMX
/* Display the system time in the text-constant
/* control #TC-1 and update the time (= timer /* control click event) every 1000 milliseconds
/* (TIMER-INTERVAL attribute).
```

### Note:

It is not recommended to write code that counts clicks in the timer click-event handler, because the click events that occur while the system is busy may get lost.

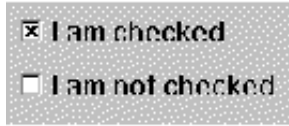
## Attributes for Timer

Attribute Name	Query	Set/Modify	In Attr. Window
CLIENT-DATA	X	X/X	
CLIENT-HANDLE	X	X/X	
CLIENT-KEY	X	X/X	
CLIENT-VALUE	X	X/X	
FIRST-CHILD	X	-/-	
LAST-CHILD	X	-/-	
PARENT	X	X/-	
PREDECESSOR	X	-/-	
SUCCESSOR	X	-/-	
SUPPRESS-CLICK-EVENT	X	X/X	
TIMER-INTERVAL	X	X/X	
TYPE	X	X/-	

## Events

Click event (may be suppressed).

## Toggle-Button Control



### Description

A toggle-button control offers the end user a checked/not checked (on/off) alternative. You use it, for example, to let the end user switch an option on or off without affecting the status of other adjacent toggle-button controls. A change in the state of a radio-button control, by contrast, will affect the state of other radio-button controls that belong to the same group.

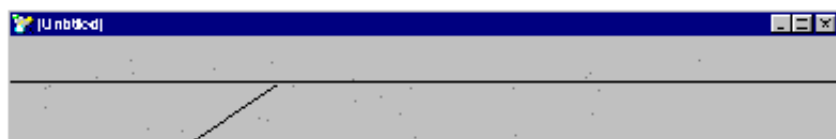
### Attributes for Toggle-Button Control

Attribute Name	Query	Set/Modify	In Attr. Window
ACCELERATOR	X	X/X	X
BACKGROUND-COLOUR-NAME	X	X/X	X
BACKGROUND-COLOUR-VALUE	X	X/X	X
CHECKED	X	X/X	X
CLIENT-DATA	X	X/X	
CLIENT-HANDLE	X	X/X	
CLIENT-KEY	X	X/X	
CLIENT-VALUE	X	X/X	
CONTEXT-MENU	X	X/X	X
DIL-TEXT	X	X/X	X
ENABLED	X	X/X	X
FIRST-CHILD	X	-/-	
FOLLOWS	X	X/X	
FONT-HANDLE	X	X/X	X
FOREGROUND-COLOUR-NAME	X	X/X	X
FOREGROUND-COLOUR-VALUE	X	X/X	X
HELP-ID	X	X/X	X
LAST-CHILD	X	-/-	
MODIFIED	X	-/X	
MODIFIED-SUCCESSOR	X	-/-	
PARENT	X	X/-	
PREDECESSOR	X	-/-	
RECTANGLE-H	X	X/X	X
RECTANGLE-W	X	X/X	X
RECTANGLE-X	X	X/X	X
RECTANGLE-Y	X	X/X	X
STRING	X	X/X	X
SUCCESSOR	X	-/-	
SUPPRESS-CLICK-EVENT	X	X/X	
TYPE	X	X/-	
VISIBLE	X	X/X	X

## Events

Click event (may be suppressed).

## Tool Bar



This is a toolbar control without any toolbar items.

## Description

A tool bar is similar to a menu bar. It consists of horizontally or vertically grouped tool-bar items with a bitmap on each of them, providing quick access to the most frequently needed program functions. By default, it is located at the top of the dialog, directly below the menu bar. It can also be located at the bottom, the left or the right of the dialog window.

A tool bar only becomes visible within a dialog when its handle value is assigned to the dialog's TOOLBAR-HANDLE attribute and HAS-TOOLBAR is TRUE for the dialog. When you create the tool bar with the dialog editor, the dialog editor does this automatically. For MDI child windows, the tool bar is displayed at the top of the MDI frame window rather than in the child window itself. Any time another MDI child window is activated, the tool bar changes to reflect the tool bar defined for the particular MDI child window.

You can determine the size of the tool-bar items using the ITEM-W and ITEM-H attributes. The distance between the items and the tool bar's border is determined with the MARGIN-X and MARGIN-Y attributes. If you set the tool bar's style to "w", the items that would exceed the tool bar's width are wrapped around.

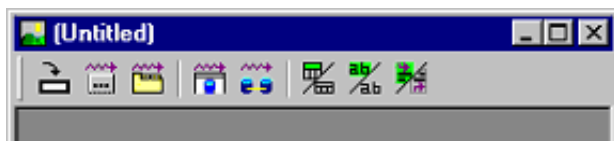
## Attributes for Tool Bar

Attribute Name	Query	Set/Modify	In Attr. Window
CLIENT-DATA	X	X/X	
CLIENT-HANDLE	X	X/X	
CLIENT-KEY	X	X/X	X
CLIENT-VALUE	X	X/X	
FIRST-CHILD	X	-/-	
ITEM-H	X	X/X	X
ITEM-W	X	X/X	X
LAST-CHILD	X	-/-	
MARGIN-X	X	X/X	X
MARGIN-Y	X	X/X	X
PARENT	X	X/-	
PREDECESSOR	X	-/-	
STYLE	X	X/X	X
SUCCESSOR	X	-/-	
TYPE	X	X/-	

## **Events**

This dialog element does not create events.

## Tool-Bar Control



### Description

A tool-bar control is an alternative to the traditional tool bar described above. It, too, can contain one or more tool-bar items used to trigger specific application functionality. However, the tool-bar control offers a range of advanced features which the traditional Natural tool bars do not support. These new features include a Windows-like appearance with (optionally) flat buttons, the ability to specify that the tool bar should be "dockable" (see below), the ability to embed other dialog elements (such as selection-boxes), and the ability to define tool tips for the tool-bar items.

A tool-bar control is dockable if its DRAGGABLE attribute is set to TRUE when the control is created. Dockable tool bars have the advantage that they can be repositioned by the user by tearing off the tool-bar control via clicking on the gripper (see illustration above), a separator or any part of its background area and dragging it with the primary mouse button held down. When the tool-bar control is dropped, it will either be snapped into position at the side of the dialog ("docked"), or "floated" in its own window. Whether the control is floated or docked depends on where the control was dropped, whether the dialog and tool-bar control both allow docking on the target side and whether the CTRL key is being held down. In the latter case, the tool bar will be floated regardless of its drop position. The border width of the drag rectangle informs the user as to whether the tool-bar control will be floated (thick border) or docked (thin border). The dragging process can be aborted at any time by pressing the ESC key.

A further major benefit of using tool-bar controls instead of the traditional tool bars is that multiple tool-bar controls can be used within a dialog, which can be independently docked or floated as required. The layout can be preserved on a per-user basis between sessions by activating the "save layout" option in the Dialog Attributes window in the dialog editor.

You can determine the size of the tool-bar items using the ITEM-W and ITEM-H attributes. The distance between the items and the tool-bar control's border is determined with the MARGIN-X and MARGIN-Y attributes.

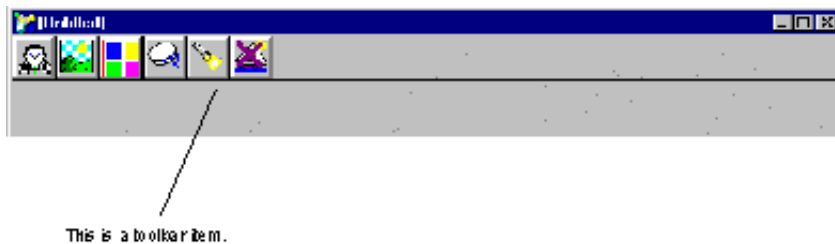
### Attributes for Tool-Bar Control

Attribute Name	Query	Set/Modify	In Attr. Window
BAR-ID	X	X/-	
CLIENT-DATA	X	X/X	
CLIENT-HANDLE	X	X/X	
CLIENT-KEY	X	X/X	
CLIENT-VALUE	X	X/X	X
CONTEXT-MENU	X	X/X	X
DOCKING	X	X/X	X
DRAGGABLE	X	X/-	X
ENABLED	X	X/X	X
FIRST-CHILD	X	-/-	
FOLLOWS	X	X/X	
HAS-DIL	X	X/X	X
HAS-TOOLTIP	X	X/X	X
ITEM-H	X	X/-	X
ITEM-W	X	X/-	X
LAST-CHILD	X	-/-	
LOCATION	X	X/X	X
MARGIN-X	X	X/X	X
MARGIN-Y	X	X/X	X
PARENT	X	X/-	
PREDECESSOR	X	-/-	
RECTANGLE-H	X	X/-	X
RECTANGLE-W	X	X/-	X
RECTANGLE-X	X	X/X	X
RECTANGLE-Y	X	X/X	X
STRING	X	X/X	X
STYLE	X	X/-	X
SUCCESSOR	X	X/-	
TYPE	X	X/-	
VISIBLE	X	X/X	X

## Events

This dialog element does not create events.

## Tool-Bar Item



### Description

A tool-bar item is an item inside a tool bar. A tool-bar item is similar to a push button with a bitmap displayed on it. In most cases, a tool-bar item will serve as an alias for a menu item. So, instead of programming again the same click-event handler code as for the menu item, you can use the tool-bar item's SAME-AS attribute and assign the corresponding menu item handle value. If the tool-bar item is clicked, the event handler of the corresponding menu item is triggered. If you do not use the SAME-AS attribute, you must specify the event handler code with the tool-bar item.

You can determine the size of the tool-bar items using the parent tool bar's ITEM-W and ITEM-H attributes.

#### **Note:**

For performance reasons, you should not use more than 16 colors in tool bar icon bitmaps; all bitmaps should share the same palette of colors.

### Attributes for Tool-Bar Item

Attribute Name	Query	Set/Modify	In Attr. Window
ACCELERATOR	X	X/X	X
BACKGROUND-COLOUR-NAME	X	X/X	X
BACKGROUND-COLOUR-VALUE	X	X/X	X
BITMAP-FILE-NAME	X	X/X	X
CHECKED	X	X/X	X
CLIENT-DATA	X	X/X	
CLIENT-HANDLE	X	X/X	
CLIENT-KEY	X	X/X	
CLIENT-VALUE	X	X/X	
DIL-TEXT	X	X/X	X
ENABLED	X	X/X	X
FIRST-CHILD	X	-/-	
LAST-CHILD	X	-/-	
MENU-ITEM-TYPE	X	X/-	X
PARENT	X	X/-	
PREDECESSOR	X	-/-	
SAME-AS	X	X/X	X
SHARED	X	X/X	X
STYLE	X	X/-	X
SUCCESSOR	X	X/-	
TOOLTIP	X	X/X	X
TYPE	X	X/-	
VISIBLE	X	X/X	X

## Events

Click event.